

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПОЧИВНИ ДНИ

Първо прочитаме началната дата **startDate** и крайната дата **endDate**. Четем датите като поредица от **три цели числа** (ден, месец и година), като пропускаме разделителя (точка).

Най-лесно можем да решим задачата като **преминем през всички дати от startDate до endDate** включително и **преброим** колко от тях са събота или неделя. Всички дати между най-ранната възможна и най-късната възможна са около 73 200 (200 години * 366 дни). Толкова проверки ще са достатъчно бързи и няма нужда да търсим по-ефективен алгоритъм.

Понеже, не знаем какъв ден от седмицата е началната дата (а това не е много лесно да се пресметне), ще стартираме от **1.01.1900 г.** (който по условие е **понеделник**).

Алгоритъм

Преминаваме през всички дати **d** в периода от **1.01.1900 г.** (понеделник) до **endDate** и проверяваме за всяка от тях дали **d ≥ startDate** (т.е. дали е в сила **startDate ≤ d ≤ endDate**) и ако **d** е почивен ден (събота или неделя), увеличаваме брояча за почивни дни (който в началото стартира от 0).

Изчисляване на следващ ден

Как преминаваме от една дата към следващата? Да разгледаме календара:

- 1.01.1900 – понеделник, работен ден
- 2.01.1900 – вторник, работен ден
- ...
- 6.01.1900 – събота, почивен ден
- 7.01.1900 – неделя, почивен ден
- 8.01.1900 – понеделник, почивен ден
- ...
- 31.01.1900 – сряда, работен ден
- 1.02.1900 – четвъртък, работен ден
- ...
- 28.02.1900 – сряда, работен ден
- 1.03.1900 – четвъртък, работен ден
- ...

За да **преминем от една дата към следващата** (например от 31.12.2016 г. към 1.01.2017 г.) първо увеличаваме деня с 1. Ако денят прехвърли броя дни в месеца, преминаваме към първия ден в следващия месец (ден = 1; месец++). Ако и месецът прехвърли броя месеци в годината (т.е. месец > 12), преминаваме към първия месец от следващата година (месец = 1; година++).

Изчисляване на ден от седмицата

Първоначално започваме от най-ранната възможна дата за тази задача: **1.01.1900 г.** Тази дата по условие е **понеделник** (може да се провери в календара).

При всяко преминаване към следващ ден, сменяме и деня от седмицата (**dayOfWeek**) към следващия: понеделник → вторник, вторник → сряда, ..., неделя → понеделник. Ако номерираме дните от седмицата от **0** (понеделник) до **6** (неделя), то можем да получим следващия ден от седмицата по следната формула: **dayOfWeek = (dayOfWeek + 1) % 7**.

Остава да извършим още две пресмятания: проверка дали една дата е преди друга и изчисляване на броя дни в месеца (по месец и години).

Брой дни в месеца

Броят дни в месеца (по номер на месец и година) пресмятаме като съобразим, че:

- Месеци 1 (януари), 3 (март), 5 (май), 7 (юли), 8 (август), 10 (октомври) и 12 (декември) имат по **31 дни**.
- Месеци 4 (април), 6 (юни), 9 (септември) и 11 (ноември) имат по **30 дни**.
- Месец 2 (февруари) има **29 дни** при високосна година и **28 дни** в останалите години.
- **Високосна** е всяка година, която:
 - се дели на 4 и не се дели на 100 (например 2016)
 - се дели на 400 (например 2000)

Сравнение на дати

Как да проверим дали една дата е преди друга? Първо сравняваме **годините**, после **месеците** (ако годините съвпадат) и накрая **дните** (ако годините и месеците съвпадат). Ако дните, месеците и годините съвпадат, имаме съпадащи дати, иначе едната от датите е по-ранна. Можем да напишем функция, която връща **-1**, **0** или **1** според това коя от две подадени дати е по-ранна. С тази функция лесно установяваме дали за дадена дата **d** е в сила $\text{startDate} \leq d \leq \text{endDate}$.

Външни библиотеки за дати

В езика **C++** и неговата стандартна библиотека няма вградени средства за изчисления с дати, така че остава да напишем сами логиката за сравнение на дати, изчисление на ден от седмицата, преминаване към следващ ден и подобни. Функциите от **time.h** могат да помогнат, но не са твърде удобни.

В езици като **C#** и **Java** тази задача може да се реши значително по-лесно. Структурата [DateTime](#) в **C#** и класът [LocalDate](#) в **Java** предоставят лесни за използване изчисления с дати (сравнение, ден от седмицата, прибавяне / извеждане на време и други).

Автор: Светлин Наков